

carloop-library

<https://github.com/carloop/carloop-library>

Make apps for your car using signals from OBD-II, CAN and GPS. Publish data online using the Particle platform.

The [hardware](#) and [software](#) is all open source!

Don't have one yet? [Buy a Carloop here](#)

Have questions? [Ask the Carloop community](#).

Want to learn about Car Hacking? [Read this book](#)

Usage

```
#include "carloop/carloop.h"
Carloop carloop;

void setup()
{
  carloop.begin();
}

void loop()
{
  carloop.update();
  CANMessage message;
  if(carloop.can().receive(message)) {
    // Do awesome things with message!
  }
}
```

See complete examples in [the examples directory](#).

Getting started

[Follow these instructions on the community](#) to create your first application using the Carloop library.

Learn about programming with the Particle platform at <https://docs.particle.io>

If you want to add a GPS, we recommend [the GP-735](#)

Documentation

Carloop object

Create a global `Carloop` object and specify the hardware revision (`CarloopRevision2`). Find this information on the Carloop circuit board (R2.1 means revision 2).

Call `begin()` in `setup()` and `update()` in `loop()` .

```
Carloop carloop;
```

```
void setup() {  
  carloop.begin();  
}
```

```
void loop() {  
  carloop.update();  
}
```

can()

Read and write raw CAN messages.

```
void setup() {  
  carloop.begin();  
  Serial.begin(9600);  
}  
  
void loop() {  
  carloop.update();  
  CANMessage message;  
  while(carloop.can().receive(message))  
  {  
    Serial.printf("ID %03x len %d", message.id, message.length);  
    for(int i = 0; i < message.length; i++) {  
      Serial.printf("%02x ", message.data[i]);  
    }  
    Serial.print("\n");  
  }  
}
```

[See the documentation for the Particle CAN driver for more options.](#)

gps()

If you have a GPS connected to the Carloop, you access GPS function through the TinyGPS object with `carloop.gps()` . To ensure that the GPS location doesn't update in the middle of reading it, surround the block where you extract the GPS data in a `WITH_LOCK` block.

```
WITH_LOCK(carloop.gps()) {  
  float lat = carloop.gps().location.lat();
```

```
float lng = carloop.gps().location.lng();
bool gpsValid = carloop.gps().location.isValid();
}
```

See the [TinyGPS library](#) for detailed examples.

battery()

The voltage of the car battery. Make sure you call `carloop.update()` in the `loop()` function to update the battery voltage.

```
float battVoltage = carloop.battery();
```

TODO

- OBD PID decoder. `carloop["engine_speed"]` to get engine speed.
- OBD fault code reader.

License

Copyright 2016-2017 1000 Tools, Inc. Distributed under the MIT license. See [LICENSE](#) for details.
